Paths to stable allocations

Ágnes Cseh*

November 12, 2012

Abstract

The stable allocation problem is one of the broadest extensions of the stable marriage problem on bipartite graphs. In this paper, we investigate the case of uncoordinated allocation processes. In this setting, a feasible allocation is given and the aim is to reach a stable allocation by raising the value of the allocation along blocking edges. This can be seen as a generalization of the paths to stability problems in the matching case.

Here we describe an algorithm that yields a stable allocation in finite time for all rational input data. With this we also show that random best response strategies converge to a stable allocation with probability one.

Keywords. stable matching, allocations, paths to stability, best response strategy

1 Stable allocations

The theory of stable matchings has been investigated for decades now. Gale and Shapley [3] introduced the notion of stability on their well-known stable marriage instance. This instance consists of a bipartite graph, where the color classes symbolize men and women, respectively. Each participant sets up a preference list of their acquaintances of the opposite gender. A set of marriages is stable, if no pair blocks it. A blocking pair is an unmarried pair such that the man is unmarried or he prefers the woman to his current wife and vice versa, the woman is unmarried or she prefers the man to her current husband. The widely known Gale-Shapley algorithm was the first proof for the existence of stable marriage schemes.

The marriage problem has been extended in several directions. One of them deals with allocations. In this problem, vertices have quota $q : V(G) \to \mathbb{R}_{\geq 0}$, and edges have capacity $c : E(G) \to \mathbb{R}_{\geq 0}$. In order to avoid confusion caused by terms associated to the marriage model, we call the vertices of the first color class jobs, while the remaining vertices are the machines. The quota of a job is the time required to spend on this job in total in order to finish it. The maximal time spent working is the quota for each machine. In additional, the machines have a time limit spent on a specific job, this is modeled by the capacity of the edges. A feasible allocation is a set of contracts where no machine is overwhelmed and no job is worked on after it has been finished.

Definition 1.1 (allocation)**.** *Function $x : E(G) \to \mathbb{R}_{\geq 0}$ is called an* allocation *if both of the following hold for every edge jm and vertex v of G:*

*TU Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany, e-mail: `cseh@math.tu-berlin.de`

1. $x(jm) \leq c(jm)$*;*

2. $\sum_{jm \in \delta(v)} x(jm) \leq q(v)$*.*

To define stability we need preference lists as well. All jobs rank their edges strictly and all machines set up a list of the jobs they are able to work on. We denote $\sum_{e \in \delta(v)} x(e)$ by $x(v)$.

Definition 1.2 (stable allocation)**.** *An allocation x is* blocked *by an edge jm if all of the following properties hold:*

1. $x(jm) < c(jm)$*;*

2. j prefers jm to its worst edge in x or $x(j) < q(j)$;

3. m prefers jm to its worst edge in x or $x(m) < q(m)$.

In other words, edge jm is blocking if it is unsaturated and neither end vertices of jm could fill up its quota with at least as good edges as jm. Allocation x is *stable* if none of the edges of G block it.

Baïou and Balinski [2] prove that stable allocations always exist. They also gave two algorithms for finding them, an extended version of the Gale-Shapley algorithm and their inductive algorithm. The worst case running time of the first solution is exponential, but the latter one runs in strongly polynomial time.

2 Uncoordinated markets

Central planning is needed in order to produce a stable solution with either of the known algorithms. In many real-life situations such a coordination is not available. The agents play their selfish strategy, trying to reach the best possible solution. Note that stability is a natural property of those markets. A stable allocations seems to be the best reachable solution for each participant, because they cannot find any partnership that could improve their own position.

The study of uncoordinated matching processes has a long history. Two basically different concepts have been studied in the topic: better and best response dynamics. According to the *better response dynamics*, a blocking edge e is chosen and added to current subset of edges. In order to preserve the matching property, one or two edges may be deleted at the two end vertices of e, depending on whether they are matched or not. *Best response dynamics* differ from better response dynamics only at one point. The blocking edge chosen has the best ranking on the preference list of its end vertex on the active side. The active color class is fix. This means that the active vertices always improve their position to the currently reachable situation.

The first question about uncoordinated two-sided markets was raised by Knuth [4] in 1976. He also gave an example for a matching problem, where better response dynamics cycle. More than a decade later came up Roth and Vande Vate with the next result in the topic. They showed that the random better response dynamics converge to a stable matching with probability one. Similar results for the best response dynamics have been published in 2008 by Ackermann [1] et al. They also proved that the convergence time is exponential in both cases.

3 Uncoordinated allocations

In this chapter, we study the case of allocations on an uncoordinated market. A feasible allocation x is given at the beginning. As mentioned above, increasing x along blocking edges and decreasing it along worse edges can lead to a process that cycles. The definition of a best response strategy is not that straightforward as on the matching instance. The possible outcomes of a player are ordered lexicographically. Playing the best response strategy means not only choosing the best blocking edge to increase x on it and the worst allocated edge to decrease x on it, but also pushing the biggest possible amount of allocation from one to another.

Here we describe an algorithm that yields a stable allocation in finite time. With this we show the following theorem:

Theorem 3.1. *For every allocation instance with rational data there is a finite sequence of best responses that leads to a stable allocation.*

Proof. For sake of simplicity we denote the residual capacity of an edge $c(jm) - x(jm)$ by $\bar{x}(jm)$ and similarly, the residual quota of a vertex $q(v) - x(v)$ by $\bar{x}(v)$. We also refer to the set of incident edges to a vertex v with $N(v)$.

The jobs form the active side, the machines are the passive players. A blocking edge can be of two types, depending on the reason of blocking at the active side. Blocking of type I is if jm is blocking x because j prefers jm to its worst edge. Blocking of type II means that j has not filled up its quota yet, $x(j) < q(j)$. Note that the reason of the blocking property at m is not involved when defining the two groups.

The algorithm consists of two phases. The main difference between the two phases is the type of the blocking edge we take. Best-response strategy is played during the whole process. In the first phase, the jobs propose along their best-choice blocking edges of type I. We will show that this process ends with an allocation where no job has a blocking edge of type I. In the second phase, the jobs propose along their best blocking edges of type II. Later we will see that during this phase until termination, no job gets a blocking edge of type I.

Algorithm 1 Two-phased best-response algorithm

```
while ∃jm blocking of type I do
    choose one
    ImprovementI(jm)
end while
while ∃jm blocking of type II do
    choose one
    ImprovementII(jm)
end while
```

procedure IMPROVEMENTI(jm)
  if $x(j) < q(j)$ **then**
    $P := \min(x(r(j)), \bar{x}(jm), \bar{x}(m))$
    $x(r(j)) := x(r(j)) - P$
    $x(jm) := x(jm) + P$
  else
    $P := \min(x(r(j)), \bar{x}(jm), x(r(m)))$
    $x(r(j)) := x(r(j)) - P$
    $x(jm) := x(jm) + P$
    $x(r(m)) := x(r(m)) - P$
  end if
end procedure

procedure IMPROVEMENTII(jm)
  if $x(m) < q(m)$ **then**
    $P := \min(x(r(m)), \bar{x}(jm), \bar{x}(j))$
    $x(r(m)) := x(r(m)) - P$
    $x(jm) := x(jm) + P$
  else
    $P := \min(\bar{x}(jm), \bar{x}(j), \bar{x}(m))$
    $x(jm) := x(jm) + P$
  end if
end procedure

First phase. In one step, an arbitrary blocking edge of type I is chosen. Both end vertices, j and m may refuse some allocation along edges when increasing x on jm. Every job j has a refusal pointer $r(j)$ that denotes the worst allocated edge to j, if exists. Similarly, $r(m)$ stands for the worst currently allocated edge of m. A step of phase I consists of two or three operations, each along $jm, r(j)$ and possibly along $r(m)$. Two operations take place, if m has not filled up its quota yet. In this case, $x(r(j))$ will be decreased with $\min(x(r(j)), \bar{x}(jm), \bar{x}(m))$. At the same time, $x(jm)$ increases with the same amount. Depending on which expression is the minimal one, $r(j)$ gets empty or jm gets saturated or m fills up its quota. In the remaining case, if m has a full quota, three operations take place, since m has to refuse some contracts. The amount of allocation we deal with is now $P = \min(x(r(j)), \bar{x}(jm), x(r(m)))$. The allocation on the blocking edge jm will be increased with this P, on the other two edges it will be decreased with P, until one of them gets empty or saturated.

We use the following potential function in order to show that the process may not cycle:

$$\Theta(x) = \sum_{j\in J}\left([M - x(j)] \cdot \sum_{m\in N(j)} \bar{x}(jm) r_j(jm)\right)$$

In the expression above, M is a huge number, greater than the maximal quota on any job. This way neither $[M - x(j)]$, nor $\bar{x}(jm)r_j(jm)$ may be negative for any feasible allocation x. The function $\Theta(x)$ has an upper bound as well,

$$0 \leq \Theta(x) \leq |J| \cdot M \cdot \max_{j\in J} |N(j)| \cdot \max_{jm\in E} c(jm) \cdot \max_{j\in J, jm\in E} r_j(jm).$$

We will show that this function monotonically increases during the procedure. The process terminates if the amount of increment is always greater than a fix positive constant. If all data is rational, this is guaranteed.

Considering the potential function, we need to keep track on those two jobs that proposed or got refused, since the position of all other jobs remains the same. This way their term in the summation of Θ does not change.

As mentioned above, a step consists of either two or three edges changing their value in x. In the first case, when only two edges change their value in x, there is only one job j that changes its position. The allocated value of this vertex remains the same, this way the first term does not change. But the second term does, because some contracts will move from a less preferred edge to jm. This ensures the increment of $\Theta(x)$. Now, in the

second case, where three edges are involved, there is a job j that improves its position, and another job j' that loses contracts. The effect of the first change at j is just as above, the first term remains the same, the second increases. Losing contracts for j' means that both terms increase, since $x(j')$ decreases.

Second phase. When the first phase terminates, there is no blocking edge of type I. When developing the allocation along a blocking edge jm of type II, m may refuse some contracts, but j may not, since the reason of blocking is that j has not filled up its quota yet. This way we do not need the pointer $r(j)$ any more. One step consists of changes along one edge if $x(m) < q(m)$, or along two edges otherwise. If m has not filled up its quota yet, then we simply assign as many contracts to jm as possible. If m has to refuse something from j' in order to take the new offers form j, we improve m's position as long as $j'm$ gets empty or jm gets saturated or j gets its quota filled up.

First, we need to see that no step can induce a blocking edge of type I. One step in Phase II leaves all vertices but j, m and the possibly refused j' unchanged. This way if there is a blocking edge of type I, it must be incident to either of those vertices. The three cases are the following.

- $j''m$ blocks x. The situation of m got lexicographically better, this way no new blocking edge incident to m could be introduced. The existing blocking edges of type II cannot become of type I, because best-response strategy is played by j.
- jm' blocks x. The only change at j is that $x(jm)$ increases. Beforehand, j did not have a better blocking edge than jm, this is still the case.
- $j'm'$ blocks x. Similar holds for j', the only change in its neighborhood is that $x(j'm)$ decreases.

From this we see that once Phase II has started, we can never get back to Phase I. The last step ahead of us is to show that Phase II may not cycle. But this follows from the fact, that in each step exactly one machine lexicographically improves its situation. In case of a rational input, this improvement is bounded from below, this way the second phase of the algorithm terminates.

□

This algorithm also proves an important result regarding the random best response processes. Since the probability of taking a step the algorithm could take is positive for all allocations, the probability that the random process does not terminate is zero.

Theorem 3.2. *In the rational case, the random best response strategy terminates with a stable allocation with probability one.*

Polynomial time convergence cannot be shown for the general case, since best response startegies need exponential time to converge even on the matching instance.

4 Open questions

Uncoordinated processes has not been studied in the case of irrational data on the allocation model. We conjecture that the same algorithm terminates in finite time, even if irrational data is present. Further extensions of the model can also be studied from this point of view, such as stable flows.

Another possibe direction of further research is the investigation of special cases where the random procedure converges in polynomial time. Faster algorithms may also exist, hence the stable allocation problem is solvable in polynomial time.

References

[1] Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking. Uncoordinated two-sided matching markets. In Lance Fortnow, John Riedl, and Tuomas Sandholm, editors, *ACM Conference on Electronic Commerce*, pages 256–263. ACM, 2008.

[2] Mourad Baïou and Michel Balinski. The stable allocation (or ordinal transportation) problem. *Math. Oper. Res.*, 27(3):485–503, 2002.

[3] David Gale and Lloyd Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 1:9–14, 1962.

[4] Donald E. Knuth. *Mariages stables et leurs relations avec d'autres problèmes combinatoires*. Collection de la Chaire Aisenstadt. Les Presses de l'Université de Montréal, Montréal, Québec, Canada, 1976. Edition revue et corrigée, 1981. Currently available from Les Publications CRM / Centre de Recherches Mathématiques, Université de Montréal, Montréal, Québec.